SPECIFICATION

Electronic Version 1.2.8

Stylesheet Version 1.0

SPELLING CORRECTION SYSTEM AND METHOD FOR PHRASAL STRINGS USING DICTIONARY LOOPING

Background of Invention

[0001] 1. Field of the Invention

[0002] The present invention relates in general to computer-implemented spelling correction and more particularly to a spelling correction system and method for phrasal strings, such a search string used in a search engine query.

[0003] 2. Related Art

[0004] Spelling correction (or spell checking) is a widely used tool in computer applications (such as word processing applications) that verifies the correct spelling of words in written documents. A typical spelling correction technique works by encountering a word that is not in a dictionary of words (e.g. a potentially misspelled word to be checked). The potentially misspelled word is compared to words in the dictionary and the word representing the closest match is returned as the correct spelling of the word. Current spelling correction techniques were developed for spelling correction of text documents and identify words within the text by determining where spaces occur. In other words, a word within the text document is identified as a space-delimited string. In general, a space-delimited string includes a string of letters that is set apart by spaces or punctuation characters. This means that current spelling correction techniques consider the letters between the spaces (i.e. a space-delimited string) to be a "word".

[0005] There are numerous spelling correction techniques currently available. For example, some existing spell spelling correction techniques are based on a framework that allows a spelling provider to manually specify the allowable edit types and the weights that are associated with each edit. Other types of existing spelling correction techniques use context to determine the correct word and first pre-compute a set of edit-neighbors for every word within the dictionary. In these edit-neighbor techniques, a word is an edit-neighbor of another word if the word can be derived from the other word in a single edit. In this situation, an edit is defined as a single letter insertion, substitution or deletion, or a letter-pair transposition. For every word in a document, the edit-neighbor technique determines whether any edit-neighbor of a word is more likely to appear in that context than the word that was typed. All edit-neighbors of a word are assigned equal probability of having been the intended word, and then the context is used to determine which word to select. Still another type of spelling correction technique computes the probability of an intended word w given a string s by automatically deriving probabilities for all edits. Similar to the edit-neighbor technique, this probabilistic technique pre-computes a set of allowable edits and only considers words w that are a single edit away from string s. Yet another spelling correction technique is a learning string edit technique that learns the probabilities for all edits, where edits are limited to single character insertion, deletion and substitution. This learning string edit technique allows for a string s to be derived from a word w by any number of edits.

[0006] All of these spelling correction techniques were developed with text document processing in mind. These techniques do not work well, however, when they are applied to strings of misspelled words. These strings are typically phrases and are known as phrasal strings. A phrasal string is a string of characters that is not necessarily space-delimited or punctuation-delimited. In other words, the string is not necessarily delimited wherever spaces occur. An error-filled phrasal string may contain unreliable spaces and many nonstandard words. For purposes of this application, nonstandard words include words that are not contained in a standard dictionary or that may have a different meaning than commonly assigned to the words. By way of example, a search engine generally has a type-in text box whereby a

phrasal string may by entered by a user. Spelling correction is important for search engines because since most queries are short even a single misspelled word or misplaced space will be extremely problematic, and can lead to highly erroneous results. Current spelling correction techniques, however, perform poorly when used for spelling correction in applications using phrasal strings. This is because current techniques perform spelling correction based on space-delimited strings and are unreliable and ineffective in situations where a user is typing a phrasal string (such as a search engine query) and accidentally inserts or omits a space.

[0007] By way of example, assume that a user inputs a line of text into a search engine query that contains the misspelled phrase "the backs treetboys". If one of the current spelling correction techniques is used to spell correct the textual line, then when the misspelled phrase is encountered the spelling correction technique will determine whether each space-delimited string (i.e. "the", "backs"and "treetboys") is in the dictionary. Because the space-delimited strings "the" and "backs" are in the dictionary, the spelling correction technique will assume them to be correctly spelled. Moreover, because the space-delimited string "treetboys" is not in the dictionary, the spelling correction technique will assume that the string is misspelled and search for the closest match in the dictionary. It is likely that a close match will not be found and the response such as "Not in Dictionary" will be returned to the user. This can be especially frustrating to a user who had intended to enter in the query box of the search engine the phrase "the backstreet boys" and accidentally added a space at a wrong location.

[0008] Another problem with using current spelling correction techniques in a search engine environment is that a dictionary is difficult to define. This is because some terms (especially proper names) used in search engine queries do not occur in standard dictionaries. Another problem is that the concept of a word is much looser in a search engine query (as compared to a word processing application) and users will often omit spaces between words. Still another problem is that case information is considerably less reliable in a search engine query than with a word processing application. This makes it difficult to know whether a potentially incorrect word not found in the dictionary is a proper name or a misspelled word. Because current

spelling correction techniques use a dictionary containing only single words (or space-delimited strings), these existing techniques are ineffective when applied to phrasal strings.

[0009] Still another problem with using current spelling correction techniques for phrasal strings is that the word dictionary used by current spelling correction techniques is essentially static while phrasal strings used in search queries are dynamic. Existing techniques use a static dictionary containing words (or space-delimited strings) for a certain language (such as English). There is no need for the dictionary to be constantly updated because new words are added to the language and old words fall out of use relatively slowly over a long period of time. Over a reasonable period of time, therefore, a dictionary for a language can be considered static. Conversely, phrasal strings used in search queries are quite dynamic, with the probability of certain phrases being used in a query varying widely from one day to the next. For example, if a major world event suddenly occurred then phrases pertaining to that event would be used often. On the other hand, a week later the phrases pertaining to that event may be old news and replaced with phrases corresponding to a more current event. In addition, search engine query terms often include jargon that does not have the standard dictionary meaning. A static dictionary cannot update itself to reflect current high-use search phrases and new meanings for words.

[0010] What is needed, therefore, is a system and method for spelling correction that is capable of operating on phrasal strings and is not limited to a single word or space-delimited text. The spelling correction system and method must be effective for spell correcting phrasal strings and such that the problem of erroneous insertions and deletions of spaces in a phrasal string can be overcome. Moreover, what is needed is a spelling correction system and a method that contains a phrasal dictionary containing entries that can be arbitrary word strings and phrases instead of merely single word entries. In addition, what is needed is a phrasal dictionary that is dynamic and is capable of being updated to include high-use phrases and words.

Summary of Invention

[0011] To overcome the limitations in the prior art as described above and other

limitations that will become apparent upon reading and understanding the present specification, the present invention includes a spelling correction system and method for spelling correction of phrasal strings. A phrasal string is a string of characters that may cross word boundaries and does not necessarily use spaces to define the limits of the string. For example, similar to a space-delimited string a phrasal string may contain single words (such as "employment") but in addition may contain multiple words (such as "employment in Arkansas"). The present invention facilitates spelling correction of entire phrasal strings rather than simply using spelling correction of space-delimited input strings. A misspelled phrasal string is segmented into a plurality of segmentations and a novel looping process through the dictionary is used to determine a segmentation having a lowest cost. A corrected segmentation corresponding to a segmentation having a lowest cost is designated as the most probable correct spelling of the phrasal string. In a preferred embodiment the dictionary of the present invention is a phrasal dictionary containing both single words and strings of words. Moreover, the phrasal dictionary of the present invention is dynamic such that phrasal strings and words used frequently are added to the dictionary. Current spelling correction techniques are based on single words and typically compare these single words to a static generic dictionary containing single word entries.

[0012] In general, the spelling correction system of the present invention includes a phrasal dictionary containing phrasal strings (single and multiple word strings) and a looping comparator that performs a novel looping process through the phrasal dictionary to correct the segmentation by determining a best match. The phrasal dictionary includes entries in a data structure and the looping comparator uses the novel looping process of the present invention to compare sub-strings of the segmentation with the dictionary entries. In a preferred embodiment, the phrasal dictionary includes a dynamic update module that provides dynamic updating of dictionary entries such that phrasal strings and words used frequently are added to the dictionary. The system of the present invention also includes a segmentation module that segments an input string into a plurality of different segmentations, with each segmentation containing contiguous sub-strings. These sub-strings are used by

the looping comparator to search the dictionary and correct the segmentation. The system of the present invention also includes a hierarchical module that provides the criteria (a set of hierarchical parameters) by which the looping comparator determines what constitutes a best match between the input string and an entry in the dictionary. These hierarchical parameters include, for example, the length of the dictionary entry and a probability of the dictionary entry given the context of neighboring words.

[0013] The present invention also includes a method for spelling correction of a phrasal string using a novel dictionary looping technique. The method includes segmenting a phrasal string into a plurality of segmentations containing sub-strings. Each sub-string is compared to corresponding dictionary entries to determine a closest match. Preferably, this comparison is performed using a dictionary looping technique. Dictionary looping allows the dictionary to be compact because the dictionary need not include all potential combinations of all possible phrases that could be encountered. The dictionary looping technique searches for a best match entry in a dictionary by looping through a dictionary data structure containing the dictionary entries. Based on this comparison of the sub-strings to dictionary entries, a cost is assigned to each of the segmentations. The dictionary entry corresponding to the segmentation having the lowest cost is designated as the corrected phrasal string.

[0014] The dictionary looping technique includes performing a number of different searches through the dictionary data structure. Each search begins at a starting node and different searches are performed by continually looping back to the starting node. An entry in the dictionary entry corresponding to the segmentation having a lowest cost is designated as the output string and by definition as the most probable correct spelling of the input phrasal string. Determining the dictionary entry having the lowest cost may also include using hierarchical parameters. These hierarchical parameters include, for example, the length of the dictionary entry and a probability of the dictionary entry. The method of the present invention also includes dynamically updating entries in the dictionary to contain the most frequently used phrasal strings entered by a user. By way of example, these current phrasal strings may be the most popular phrasal strings over a period of time.

[0015] Other aspects and advantages of the present invention as well as a more complete understanding thereof will become apparent from the following detailed description, taken in conjunction with the accompanying drawings, illustrating by way of example the principles of the invention. Moreover, it is intended that the scope of the invention be limited by the claims and not by the preceding summary or the following detailed description.

Brief Description of Drawings

[0016] The present invention can be further understood by reference to the following description and attached drawings that illustrate the preferred embodiments. Other features and advantages will be apparent from the following detailed description of the invention, taken in conjunction with the accompanying drawings, which illustrate, by way of example, the principles of the present invention.

[0017] Referring now to the drawings in which like reference numbers represent corresponding parts throughout:

[0018] FIG. 1 is an overall block diagram illustrating an implementation of the phrasal spelling correction system of the present invention and is provided for illustrative purposes only.

[0019] FIG. 2 is a general block diagram illustrating a computing platform as shown in FIG. 1 that preferably may be used to carry out the present invention.

[0020] FIG. 3 is a general block diagram illustrating components of the phrasal spelling correction system of FIG. 1 and their interaction.

[0021] FIG. 4 is a general flow diagram illustrating the general operation of the present invention.

[0022] FIG. 5 is a detailed flow diagram illustrating the detailed operation of the present invention.

[0023] FIG. 6 is a detailed flow diagram illustrating in the dictionary looping operation of the looping comparator 320 shown in FIG. 3.

[0024] FIG. 7 illustrates a working example of a data structure used by the looping comparator 320 having a phrasal dictionary stored therein and used for dictionary looping.

Detailed Description

[0025] In the following description of the invention, reference is made to the accompanying drawings, which form a part thereof, and in which is shown by way of illustration a specific example whereby the invention may be practiced. It is to be understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

[0026] I. Introduction

[0027] The present invention includes a spelling correction system and method for phrasal strings using dictionary looping. The present invention is especially useful in applications such as search engine queries that operate on phrasal strings entered by a user. In these applications the phrasal string may include a string of characters having a high percentage of words not appearing in a standard dictionary, a high spelling error rate, unreliable spacing, and certain phrases that are typed frequently. Current spelling correction techniques are ineffective in these applications because these techniques use a static dictionary and operate on space-delimited strings. In particular, existing spelling correction techniques assume that each word in a text is separated by spaces and identify individual words based on the location of these spaces. Each identified word in the space-delimited string is then compared to a single word entry in the static dictionary.

[0028] The present invention overcomes these limitations of existing spelling correction techniques. In particular, the spelling correction system and method of the present invention allows multiple possible segmentations of an input string, not just segmentation based on the location of spaces. These segmentations contain sub-strings that may contain a plurality of words and spaces. The sub-strings are compared to entries in a phrasal dictionary. The phrasal dictionary may contain not only single words but also phrases (including spaces). By comparing the sub-strings

to entries in the phrasal dictionary, the present invention is capable of resolving spelling errors due to erroneous insertion and deletion of spaces. Moreover, the present invention is also able to use contextual information as appropriate to perform spelling correction. For example, using contextual information, the present invention is able to correct the entire incorrect query, "the backstreet toyz" with "the backstreet boys" instead of merely replacing "toyz" with "toys".

[0029] Although the present invention may be used for spelling correction of both long text entries and short text entries, generally the present invention will more frequently be used to address the problem of spelling correction of short text entries. A prototypical example of this is a search engine query box. A misspelled query to a search engine can result in a user not obtaining the desired information. This can result in user frustration, lost revenue and increased cost. Spelling correction is an important tool for search engines for at least two reasons. First, the human error rate in typing search queries is much higher than the error rate in document preparation using a word processor (up to 15% by some estimates). Second, since most queries are very short, a single mistyped word or misplaced or omitted space often will result in an unproductive search.

[0030] II. General Overview

[0031] FIG. 1 is an overall block diagram illustrating an implementation of the phrasal spelling correction system 100 of the present invention and is provided for illustrative purposes only. In general, the phrasal spelling correction system 100 is a part of an application 110 (such as a search engine) residing on a computing platform 120. The application 110 also includes a user interface 130 (such as a graphical user interface) whereby a user 140 is able to interact with the application 110. The user 140 uses an input device (such as a keyboard (not shown)) to input a phrasal string (box 150) into the application 110. As shown in FIG. 1, the phrasal string is displayed in the user interface 130 (such as in an edit or query box). In this case, the intended phrasal string to be entered in the user interface 130 is "pictures of marilyn monroe". Instead, however the user 140 left out a couple of spaces, misspelled a couple of words and instead entered the phrasal string "picturesff marilynminro".

[0032] Either upon request by the user 140 or automatically, the application 110 calls the phrasal spelling correction system 100 to spell correct the phrasal string that the user 140 has typed in the user interface 130. As shown in FIG. 1, the misspelled phrasal string is received by the system 100 as "picturesff marilynminro". A traditional spelling correction technique based on space-delimited strings examines the input phrasal string and divides the misspelled string into "words" bounded by spaces. In this case, traditional spelling correction techniques would determine that the words "picturesff" and "marilynminro" are misspelled. Traditional spelling correction techniques would return "pictures" as a probable correct spelling for "picturesff" and probably no suggestion for "marilynminro". This would leave the phrase "pictures marilynminro" in the search query box. This phrase would not be an effective search query when desiring pictures of Marilyn Monroe. Thus, traditional spelling correction techniques would be ineffective in this situation.

[0033] As shown in FIG. 1, the phrasal spelling correction system 100 of the present invention segments the entire misspelled phrasal string into a plurality of segmentations containing sub-strings. These segmentations may be for any possible segmentation of the phrasal string, and are not restricted to text between spaces. In FIG. 1, the misspelled phrasal string "picturesff marilynminro" is shown segmented into segmentations that include: (1) / pictures / ff / marilyn / minro /; (2) / picturesff / marilynminro /; (3) / picturesff marilyn / minro /; and (4) / picturesff marilynminro /. It should be noted that the bolded vertical line " /" indicates where the segmentation of the misspelled phrasal string occurs. A sub-string is the string between the vertical lines.

[0034] For each segmentation, each sub-string in the segmentation is compared to entries in a phrasal dictionary. This comparison is used to determine which entry is a closest match to the sub-string being examined. For example, in the first segmentation, / pictures / ff / marilyn / minro /, the sub-strings "pictures", "ff", "marilyn" and "minro" are compared to entries in the phrasal dictionary. Although the sub-strings "pictures" and "marilyn"would be found in the phrasal dictionary, finding an entry that was close to the sub-strings "ff" or "minro" would be difficult. For the second segmentation, / picturesff / marilynminro /, the sub-string "picturesff" would

be a close match with the entry in the phrasal dictionary "pictures of". In fact, this match would be a better match than trying to match the sub-string "ff" with the dictionary entry "of". For the third segmentation, / picturesff marilyn / minro /, the sub-string "picturesff marilyn" would be a close match with the phrasal dictionary entry "pictures of marilyn". Finally, the fourth segmentation, / picturesff marilynminro /, is a close match with the phrasal dictionary entry "pictures of marilyn monroe".

[0035] As explained in detail below, the closest (or "best") match for each sub-string in a particular segmentation is determined by looping through the phrasal dictionary and comparing each sub-string to dictionary entries. The best match is defined as the match having the lowest cost, or the lowest cost of correction. In the example shown in FIG. 1, the match between the fourth segmentation, / picturesff marilynminro /, and the phrasal dictionary entry "pictures of marilyn monroe" has the lowest cost of correction and is the best match. Next, the cost of correcting each segmentation is added and the segmentation is assigned a cost. The segmentation having the lowest cost is the best segmentation, and the corrected segmentation is the designated as the most probable correct spelling of the input phrasal string. This corrected segmentation is returned (box 160) to the application 110.

[0036] In a preferred embodiment, the computing platform 120 is a computing machine (or device) in a computing environment. FIG. 2 is a general block diagram illustrating the computing platform 120 as shown in FIG. 1 that preferably may be used to carry out the present invention. FIG. 2 and the following discussion are intended to provide a brief, general description of a suitable computing environment in which the spelling correction system and method of the present invention may be implemented. Although not required, the present invention will be described in the general context of computer-executable instructions (such as program modules) being executed by a computer.

[0037]

Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with a variety of computer system configurations, including personal

computers, server computers, hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located on both local and remote computer storage media including memory storage devices.

[0038] Referring to FIG. 2, an exemplary system for implementing the present invention includes a general-purpose computing device in the form of a conventional personal computer 200, including a processing unit 202, a system memory 204, and a system bus 206 that couples various system components including the system memory 204 to the processing unit 202. The system bus 206 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. The system memory includes read only memory (ROM) 210 and random access memory (RAM) 212. A basic input/output system (BIOS) 214, containing the basic routines that help to transfer information between elements within the personal computer 200, such as during start-up, is stored in ROM 210. The personal computer 200 further includes a hard disk drive 216 for reading from and writing to a hard disk (not shown), a magnetic disk drive 218 for reading from or writing to a removable magnetic disk 220, and an optical disk drive 222 for reading from or writing to a removable optical disk 224 (such as a CD-ROM or other optical media). The hard disk drive 216, magnetic disk drive 228 and optical disk drive 222 are connected to the system bus 206 by a hard disk drive interface 226, a magnetic disk drive interface 228 and an optical disk drive interface 230, respectively. The drives and their associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules and other data for the personal computer 200.

[0039] Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 220 and a removable optical disk 224, it should be appreciated by those skilled in the art that other types of computer readable media that can store data that is accessible by a computer, such as magnetic cassettes, flash

memory cards, digital video disks, Bernoulli cartridges, random access memories (RAMs), read-only memories (ROMs), and the like, may also be used in the exemplary operating environment.

[0040] A number of program modules may be stored on the hard disk, magnetic disk 220, optical disk 224, ROM 210 or RAM 212, including an operating system 232, one or more application programs 234, other program modules 236 and program data 238. A user (not shown) may enter commands and information into the personal computer 200 through input devices such as a keyboard 240 and a pointing device 242. In addition, other input devices (not shown) may be connected to the personal computer 200 including, for example, a camera, a microphone, a joystick, a game pad, a satellite dish, a scanner, and the like. These other input devices are often connected to the processing unit 202 through a serial port interface 244 that is coupled to the system bus 206, but may be connected by other interfaces, such as a parallel port, a game port or a universal serial bus (USB). A monitor 246 or other type of display device is also connected to the system bus 206 via an interface, such as a video adapter 248. In addition to the monitor 246, personal computers typically include other peripheral output devices (not shown), such as speakers and printers.

[0041] The personal computer 200 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 250. The remote computer 250 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the personal computer 200, although only a memory storage device 252 has been illustrated in FIG. 2. The logical connections depicted in FIG. 2 include a local area network (LAN) 254 and a wide area network (WAN) 256. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

[0042]

When used in a LAN networking environment, the personal computer 200 is connected to the local network 254 through a network interface or adapter 258. When used in a WAN networking environment, the personal computer 200 typically includes a modem 260 or other means for establishing communications over the wide area

network 256, such as the Internet. The modem 260, which may be internal or external, is connected to the system bus 206 via the serial port interface 244. In a networked environment, program modules depicted relative to the personal computer 200, or portions thereof, may be stored in the remote memory storage device 252. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

[0043] III. Components of the Invention

[0044] The present invention includes a phrasal spelling correction system that inputs a phrasal string to be corrected, processes the phrasal string and outputs the most likely correct spelling of the input phrasal string. FIG. 3 is a general block diagram illustrating components of the phrasal spelling correction system 100 of FIG. 1 and their interaction. In particular, a phrasal string 300 is sent to the phrasal spelling correction system 100 and received by a segmentation module 310. The segmentation module 310 segments the phrasal string 300 into a plurality of different possible segmentations. Each of these plurality of segmentations includes sub-strings, shown in FIG. 3 as sub-string (1) to sub-string (N). The segmentation of the phrasal string is a partitioning or dividing of the phrasal string into contiguous sub-strings over the entire phrasal string. For instance, given the string "pintures ff marylinnMonroe", one segmentation is / pintures ff / marylinnMonroe / consisting of sub-string (1) = "pintures ff "and sub-string (2) = "marylinnMonroe".

[0045] For each segmentation of the phrasal string 300, the sub-strings of that segmentation are processed by a looping comparator 320. As explained in detail below, the looping comparator 320 uses dictionary looping to compare each of the sub-strings to entries in a dynamic phrasal dictionary 330. The dynamic phrasal dictionary 330 contains phrasal strings, and these phrasal strings are capable of including words, phrases, characters and spaces. Using dictionary looping, the looping comparator 320 determines the dictionary entry that most closely matches the sub-string being examined. Preferably, the looping comparator 320 includes a data structure that permits looping through phrasal strings. When the closest matching dictionary entry is found, the looping comparator 320 uses information

from a hierarchical module 340 to determine a cost for the match. The cost for each sub-string is added to arrived at a total cost for the segmentation. The matches for each sub-string are combined to produce a corrected segmentation. The corrected segmentation that corresponds to the segmentation having the lowest cost is designated as the most likely correct spelling of the phrasal string 300. This corrected segmentation is sent as an output string 350.

[0046] The phrasal spelling correction system 100 also includes a dynamic update module 360 in communication with the dynamic phrasal dictionary 330. The dynamic update module 360 receives the phrasal string 300 and determines whether all or some of the phrasal string 300 should be appended to the dynamic phrasal dictionary 330. There are several techniques that the dynamic update module 360 may use to determine whether to include the all or a portion of the phrasal string 300 in the dictionary 330. Preferably, the dynamic update module 360 determines and tracks the frequency with which a certain phrasal string has been typed and appends to the dictionary 330 those phrasal strings having the highest frequency.

[0047] IV. General Operation of the Invention

[0048] FIG. 4 is a general flow diagram illustrating the general operation of the present invention. In general, the spelling correction method determines a string representing a most probable correct spelling for a phrasal string to be spell corrected. More specifically, the spelling correction method begins by segmenting a phrasal string to be corrected (box 400). Next, a cost is determined for each of the segmentations (box 410). Finally, the segmentation having the lowest cost is identified in order to obtain the most probable correct spelling of the phrasal string (box 420).

[0049] FIG. 5 is a detailed flow diagram illustrating the detailed operation of the present invention. In particular, a phrasal string to be spell corrected is received as input (box 500). The phrasal string is segmented into a plurality of different segmentations (box 510). This segmentation of the phrasal string is performed by dividing the phrasal string into contiguous sub-strings. It should be noted that these divisions do not need to occur where spaces separate text. Next, each segmentation is corrected using dictionary looping (box 520). As discussed in greater detail below, dictionary looping

compares each segmentation with entries in a phrasal dictionary. A cost for correction is then determined for each of the plurality of segmentations (box 530). After a cost has been determined for each segmentation, the segmentation having the lowest cost is determined (box 540). The corrected segmentation having the lowest cost of correction is designated as the most probable correct string for the input phrasal string (box 550). It should be noted that the lowest-cost segmentation may be composed of more than a single corresponding entry in the phrasal dictionary.

[0050] FIG. 6 is a detailed flow diagram illustrating in the dictionary looping operation of the looping comparator 320 shown in FIG. 3. Dictionary looping allows the dictionary to be compact because the dictionary need not include all potential combinations of all possible phrases that could be encountered. In general, dictionary looping begins by receiving a segmentation to be spell corrected and corrects the segmentation using a phrasal dictionary. More specifically, a segmentation is corrected by first inputting sub-strings of the phrasal string (box 600). For each sub-string, a looping search is performed through the phrasal dictionary (box 610). Preferably, the looping search is performed using a data structure that has the phrasal dictionary stored therein, as discussed in detail below. During this search each sub-string is compared with entries in a phrasal dictionary in order to find the entry that most closely matches the sub-string (box 620). Using these closest sub-string matches a corrected segmentation is constructed (box 630). The corrected segmentation is the output (box 640).

[0051] By way of example, suppose that the input phrasal string is "picturesff marilynminro" and the segmentation is / picturesff / marilynminro /. When the sub-string "picturesff" is compared to entries in the phrasal dictionary the phrase "pictures of" will produce the closest match to the sub-string. Similarly, when the sub-string "marilynminro" is compared to entries in the phrasal dictionary the phrase "marilyn monroe" will produce the closest match. A corrected segmentation for the segmentation / picturesff / marilynminro / is constructed from the closest matches "pictures of" and "marilyn monroe" such that the corrected segmentation is / pictures of / marilyn monroe /. This corrected segmentation is then sent as output where the cost of correction is determined (box 530) and the rest of the process occurs as shown in FIG. 5.

[0052] FIG. 7 illustrates a working example of a data structure used by the looping comparator 320 having a phrasal dictionary stored therein and used for dictionary looping. The data structure shown in FIG. 7 is known as a trie 700. Branches of the trie 700 are represented as paths from a root node 710 to an end node 720. By way of example, the trie 700 shown in FIG. 7 encodes the following strings:

[0053] can

[0054] can I

[0055] do

[0056] how

[0057] how can I

[0058] how can I print

[0059] I

[0060] print

[0061] Note that using the trie 700 the string "how can I print" can be represented.

[0062] Referring to FIGS. 6 and 7, the looping comparator 320 can be used to search the trie 700 in a looping manner known as dictionary looping. Dictionary looping is a looping search that is used to find an entry in the phrasal dictionary having a closest match to a sub-string being processed. Every end node 720 on the trie 700 points back to the root node 710. This means that rather than having a space indicate the termination of one pass through the trie 700, the present invention allows looping back in the trie 700 at any place in an input phrasal string. Using this dictionary looping, the looping comparator 320 find a dictionary entry that is a closest match to the sub-string being examined.

[0063] In order to determine the cost of correcting a segmentation, matches having a longer lengths are favored. The present invention implements this by using a length-adjusted distance measure. If distance(pictures of, pictures ff) = distance

(pictures,pictures) + distance(of,ff), then little would be gained by having multiword phrases in the phrasal dictionary. The present invention avoids this problem by computing an average length-adjusted segment distance. Using standard edit distance, distance(pictures of, pictures ff)=1, distance(pictures,pictures)=0 and distance(of,ff) = 1. However, using the average length-adjusted segment distance, distance(pictures of, pictures ff)=1/11, and distance(pictures,pictures) + distance (of,ff) = 0/8 + 1/2. Thus, the longer match is rewarded and assigned a lower cost.

[0064] V. Operational Details and Working Example

[0065] The operational and technical details of the present invention will now be presented. The method of the present invention begins by selecting an input phrasal string for spelling correction. This phrasal string is then segmented into a plurality of segmentation, with each segmentation containing contiguous sub-strings. It should be noted that one or more of these sub-strings may be empty. If there is no bound on the allowable number of contiguous empty sub-strings in a segmentation, then there are an infinite number of segmentations of any string. On the other hand, if no empty sub-strings are allowed, then a string having n characters has 2^{n-1} possible segmentations. Preferably, a small number of empty sub-strings are allowed.

[0066] By way of example, if the word "dogs" is the input string the present invention will segments the word into contiguous sub-strings. If no empty sub-strings are allowed, then the number of partitions is equal to 8:

[0067]

[t1]
dogs → |d| |o| |g| |s|;
dogs → |d| |og| |s|;
dogs → |do| |g| |s|;
dogs → |d| |o| |gs|;
dogs → |do| |gs|;
dogs → |dog| |s|;
dogs → |d| |ogs|;
dogs → |dogs|.

[0068] If empty sub-strings are allowed, then the number of segmentations is infinite and includes, for example,

[0069]

[t2]
dogs → |d| |og| |s|;
dogs → |d| |ε| |og| |s|;
dogs → |d| |o| |g| |s|;
dogs → |d| |ε| |ε| |og| |s|.

[0070] where ϵ denotes an empty sub-string.

[0071] A function then is defined that takes a string and a dictionary as inputs and returns a cost. Mathematically, this function may be defined as least_cost (x,D), where string x and a dictionary D are inputs. A number n is returned, with n being the cost for the string x. The number n is represented by the equation:

[0072]

$$\min_{w \in D} Dst(x, w)$$

[0073] for some predefined distance measure Dst, and is the smallest cost between x and some string w in the dictionary D. The function least_cost_string(x,D) is defined as:

[0074]

$$\arg\min_{w \in D} Dst(x, w)$$

[0075] The present invention may be implemented as follows. Assume an input string S consisting of a sequence of characters (including spaces) c1 c2 ... cn, and a dictionary D. For each segmentation (or partition P) of the input string S, a score is assigned:

[0076]

Score(P,S) = $$\sum_{p \in P} least\_cost(p, D)$$

[0077] For a partition P' for which Score(P',S) is smallest

[0078] For i = 1 to number of partitions in P'

[0079] If least_cost_string(p_i,D) = p_i then do nothing

[0080] Else suggest least_cost_string(p_i,D) as a correction for p_i

[0081] This algorithm works regardless of whether the dictionary D contains multi-word entries. Instead of suggesting a single correction, the system could also return the x most likely corrections by returning the x strings in D that are closest to p_i instead of just returning the single closest string.

[0082] Two working examples will now be presented. It should be noted that these

working examples are exemplary and many other implementations of the present invention are possible. In these working examples the dictionary D only contains single words. In a first working example, the user types the phrase: "pict uresof monstar trucks". One possible segmentation of this string would be:

[0083] / pict ures / of / monstar / trucks /

[0084] The score for this particular partition would be:

[0085] least_cost("pict ures",D)+ cost of adding a space + least_cost("of",D) + least_cost ("monstar",D) + least_cost("trucks",D)

[0086] Once this is chosen as the segmentation having the lowest score, then the corresponding corrections are:

[0087] least_cost_string("pict ures",D) = "pictures";

[0088] least_cost_string("of",D) = of;

[0089] least_cost_string("monstar",D) = "monster"; and

[0090] least_cost_string("trucks",D) = trucks.

[0091] Therefore, the present invention would suggest that the original user input of "pict uresof monstar trucks"be rewritten as "pictures of monster trucks".

[0092] In another example, the user has entered the above string as "pict uresof monstair trucks". It is unlikely that least cost_string("monstair",D) will be "monster", so it is likely that this misspelled word will have spelling correction incorrectly performed. However, if the phrase "monster trucks" appears in the phrasal dictionary of the present invention, then the best scoring (i.e. lowest cost) segmentation would likely be:

[0093] / pict ures / of / monstair trucks /

[0094] with the corresponding corrections:

[0095] least_cost_string("pict ures",D) = "pictures";

[0096] least_cost_string("of",D) = of;

[0097] least_cost_string("monstair trucks",D) = "monster trucks".

[0098] The partition P can be found more efficiently using numerous efficiency techniques for dictionary searching. These searching efficiency techniques are well known to those having ordinary skill in the art. By way of example, one such searching efficiency technique is a beam search. The preferred implementation of performing the searching, however, is to perform looping through a trie. This type of dictionary searching is explained in detail above.

[0099] The foregoing description of the preferred embodiments of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description of the invention, but rather by the claims appended hereto.